IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	O' Doherty, M
Serial No.	:	09/520,853
Filed	:	March 7, 2000
For	:	Java Enhanced SIP
Examiner	:	Jean Gilles, Jude
Art Unit	:	2143
Customer number	:	23644
Confirmation No.	:	8975
Attorney Docket No.	:	920584-906003

AMENDED APPEAL BRIEF

Honorable Director of Patents and Trademarks
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This appeal is from the Examiner's final Office Action mailed December 12, 2007 in which pending Claims 1, 2, 5-12, 14-18, 20-26 and 34 were rejected and Claims 13 and 19 were objected to. A timely Notice of Appeal was filed February 11, 2008 with the required fee.

The brief was originally filed along with the required $510.00 fee pursuant to 37 C. F. R. § 41.20(b)(2) on April 11, 2008.

(i) Real Party in Interest

This application is assigned to Nortel Networks Limited. The assignments are recorded at Reel 010633, Frame 0187.

(ii) Related Appeals and Interferences

There are no known related appeals or interferences which have any relation to this appeal.

(iii) Status of Claims

This application was filed with claims 1 to 33. In the responses of December 23, 2004, June 10, 2004, June 4, 2003, January 20, 2003, May 15, 2006, and August 27, 2007 claims 1, 3, 8, 10, 13, 15-24, 26, 27, and 30-33 were each amended at least once.

Additionally, in the responses of May 15, 2006, September 22, 2006 and August 27, 2007, new Claim 34 was added, Claims 1 to 26 and 34 were elected and Claims 3 and 4 were cancelled without prejudice, respectively.

The remaining claims are as originally filed. Claim 27-33 are withdrawn from consideration.

It is the rejection of claims 1, 2, 5-26 and 34 as set forth in the final Office Action mailed December 12, 2007 that is appealed. All the pending claims 1, 2, 5-26 and 34 are set forth in the Claims Appendix and are the claims appealed.

(iv) Status of Amendments

No amendment or response was filed following the December 12, 2007 final Office Action.

(v) Summary of Claimed Subject Matter

Claim 1

This relates to a method of transferring computer software code between a first and a second node in a communications network, each of the nodes including a SIP client. This type of network is illustrated in Figure 1 and described from page 9 line 16 onwards.

In the method computer software code is stored within a SIP message as shown in step 20 of Figure 2 and the associated description on page 10 line 6. An example of a SIP message including computer software code is described on page 13 lines 15 to 22 and is illustrated in Figure 5.

The SIP message including the computer software code is sent from the first SIP client associated with the first node to the second SIP client associated with the second node as illustrated in Step 21 of Figure 2. Figure 3 also illustrates the SIP message (36) being sent from a first SIP client (SIP client A 30) to a second SIP client (SIP client B 31). This is described on page 10 line 26.

The second node including the second SIP client (SIP client B) executes the computer software code when it has received the SIP message as described on page 10 line 26 to page 11 line 2 of the description as filed.

Claim 20

This claim relates to a node, such as those illustrated in Figure 1, within a communications network. Such a node is described on page 9 lines 16 to 26.

The node includes a SIP client (11 in Figure 1 or 30 and 31 in Figure 2), an input arranged to receive SIP messages and a processor arranged to extract and execute computer software code from a received SIP message.

Claim 24

Claim 24 relates to a computer program arranged to control a node, such as those illustrated in Figure 1, within a communications network. The node includes a SIP client and a processor and an example of such a node is described on page 9 lines 16 to 26.

The computer program is executed on the processor to control the node and causes software code stored within a SIP message to be executed by the processor when the SIP message is received by the SIP client as described on page 10 line 26 to page 11 line 2.

(vi) Grounds of Rejection To Be Reviewed on Appeal

There is 1 rejection at issue:

1. the rejection of claims 1, 2, 5-12, 14-18, 20-26 and 34 under 35 USC § 103(a) as being unpatentable over ChaiTime ("ChaiTime: A System for Rapid Creation of Portable Next-Generation Telephony Services Using Third-Party Software Components) and Schuster (US Patent No. 6,567,399).

(vii) Argument

The Examiner maintains that Claims 1, 2, 5-12, 14-18, 20-26 and 34 are not patentable over ChaiTime in view of Schuster. Applicant respectfully disagrees.

As discussed during prosecution to date, ChaiTime does not disclose a method including the step of "storing computer software code in a SIP message" or the step of "sending the SIP message and computer software code from the first SIP client associated with the first node to the second SIP client associated with the second node" as recited in Claim 1.

Rather, Applicant submits that ChaiTime describes a method of setting up a call using conventional SIP messages. If the call requires a user terminal to have software installed the user terminal can be sent a message, "indicating a service provider, reachable through a URL, which may be able to supply the resource" (Page 27 second column, last paragraph). An example of a message flow to set up a call is illustrated in Figure 6. Nowhere does ChaiTime disclose or even suggest including the computer software code for an application within a SIP message.

The Examiner states that the feature of storing computer software code in a SIP message is shown in Figure 3 on page 25. However, Applicant respectfully disagrees. Figure 3 illustrates a protocol stack present on each of the nodes within the system, including the SIP applications which would be required for SIP to be used for call signalling in a network.

In view of the foregoing Applicant submits that ChaiTime does not disclose the step of "storing computer software code in a SIP message" as recited in Claim 1.

As discussed above ChaiTime only discloses inserting a URL into a message to indicate to an endpoint where computer software code may be accessed. Applicant submits that a URL cannot be considered to be computer software code. A

URL is simply a location address on a network comprising no more than a set of characters. It is therefore submitted that a URL does not and could not perform any function other than provide a location address. Furthermore a URL does not have any of the well recognized characters of "computer software code".

As discussed in the response to the Advisory Action dated November 20, 2003, a URL (as defined in RFC2396) is a subset of the set of uniform resource indicators (URI) all of which "provide a simple and extensible means for identifying a resource" (see section 1.0 RFC 2396). RFC 2396 continues by explaining that "having identified a resource, a system may perform a variety of operations on the resource, as might be characterized by such words as "access", "update", "replace", or "find attributes"". Thus a URL is never executed as required by Claim 1. Instead, a URL may be used to point to a resource which may itself be executed (see for example claim 4 of the present application).

Applicant therefore submits that ChaiTime also does not disclose or suggest "sending the SIP message and computer software code from the first SIP client associated with the first node to the second SIP client associated with the second node" as recited in Claim 1.

The Examiner cites Schuster as teaching "a method for storing code in a SIP message". Applicant respectfully disagrees.

Schuster describes a method for enabling high-fidelity voice/audio communication. This is done by a high-fidelity SLIC (HSLIC) card being provided with a SIP client (column 8 lines 17 to 18). The SIP client acts in the conventional manner of formulating "an INVITE request to the called party... [typically containing]... a session description that provides the called party with enough information to join the call" (Column 8 lines 44 to 51). If the contacted agent accepts the call "it returns an OKAY message" (Column 8 lines 57 to 58). "The caller's HSLIC

SIP agent then sends an ACK message to the callee's HSLIC SIP agent" (Column 8 lines 59-60).

According to Schuster this method may be implemented in "software modules as a set of computer executable software instructions" (Column 8 line 67 to Column 9 line 1). These instructions are stored in memory and executed by the CPU. However, nowhere does Schuster disclose that any SIP messages are adapted to include computer software code as is presently claimed.

Applicant therefore submits that Schuster does not disclose the steps of "storing computer software code in a SIP message" and "sending the SIP message and computer software code from the first SIP client associated with the first node to the second SIP client associated with the second node" as recited in Claim 1.

Thus, Applicant submits that Claim 1 is patentable over the combined teachings of ChaiTime in view of Schuster.

Claims 20 and 24 recite the features of "a processor arranged to extract and execute computer software code from a received SIP message" and a computer program that, when executed upon a node "when executed on the processor such that when a SIP message is received by the SIP client, which contains computer software code, the software code is executed by the processor". As discussed above neither ChaiTime nor Schuster discloses or even suggests inserting computer software code from a received SIP message. Therefore neither ChaiTime nor Schuster can be said to disclose or even suggest extracting and executing the computer software code.

Applicant therefore submits that Claims 20 and 24 are patentable over ChaiTime in view of Schuster.

In connection with claim 2, the Examiner refers to the JAVA telephony API. It is clear from Figure 4 for example that the JAVA code is not added to a SIP message. It is merely used to control the underlying session initiation protocol which as explained above may be SIP or H232 or any equivalent. At the bottom of column 1 of page 27 it is explained that "the state machine for the JTAPI connection object has been extended in order to support media negotiation during call set up". Thus the JTAPI controls SIP, it is not carried in SIP messages. This rejection is therefore submitted to be in error.

In connection with claim 5, it is noted that the JTAPI software is not stored in SIP messages. The Examiner's comments in relation to this claim are not understood. This rejection also is submitted to be in error.

In connection with claim 6, the Examiner has failed to link the JAVA applets mentioned in the portion referred to in any way with SIP. These Java applets are not stored in a SIP message in ChaiTime. This rejection is therefore in error.

The Examiner has not raised any argument against claim 7.

In connection with claim 8 there is no discussion in ChaiTime of Java mobile agents. This rejection is thus in error.

In connection with claim 10, since ChaiTime does not disclose the adding of computer software code to a SIP message in any way, it certainly does not disclose the addition of the software code to a SIP message body. This rejection is therefore in error.

In connection with claim 11, since ChaiTime does not disclose the addition of computer software code to a SIP message, it cannot disclose the addition of a header. Such an indicator in a header of the SIP message is not therefore implicitly or explicitly disclosed in ChaiTime. This rejection is thus in error.

In connection with claim 13, since there is no indicator or a SIP message containing a computer software code in ChaiTime, the features of claim 13 cannot be shown. This rejection is in error.

In connection with claim 14, since there is no computer software code in terms of claim 1 disclosed in ChaiTime, the features of claim 14 cannot be shown. This rejection is therefore in error.

Similar comments apply to the remaining dependent claims and these rejections are thus submitted to be in error.

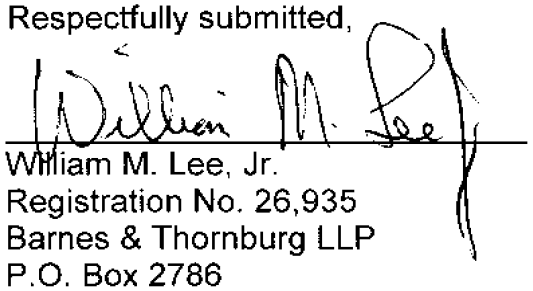
The rejections of the remaining independent claims is also submitted to be in error for similar reasons.

In connection with all the dependent claims, it is noted that these claims are allowable at least by virtue of their dependency.

In view of the above, reversal of the Examiner is urged.

July 15, 2008

Respectfully submitted,



William M. Lee, Jr.
Registration No. 26,935
Barnes & Thornburg LLP
P.O. Box 2786
Chicago, Illinois 60690-2786
(312) 214-4800
(312) 759-5646 (fax)

Claims Appendix

1. A method of transferring computer software code between a first and a second node in a communications network, each of said nodes comprising a SIP client, said method comprising the steps of:-

(i) storing computer software code in a SIP message;

(ii) sending the SIP message and computer software code from the first SIP client associated with the first node to the second SIP client associated with the second node; and

(iii) executing the computer software code using the second node.

2. A method as claimed in claim 1 wherein said computer software code is added to the SIP message.

3. (cancelled)

4. (cancelled)

5. A method as claimed in claim 1 wherein said computer software code comprises Java byte code.

6. A method as claimed in claim 1 wherein said computer software code comprises one or more Java applets.

7. A method as claimed in claim 1 wherein said computer software code comprises one or more mobile automated software agents.

8. A method as claimed in claim 7 wherein said mobile automated software agents are Java mobile agents.

9. A method as claimed in claim 1 wherein said second node comprises a Java virtual machine.

10. A method as claimed in claim 2 wherein the computer software code is added to the body of the SIP message.

11. A method as claimed in claim 1 which further comprises adding an indicator to a header of the SIP message in order to indicate the presence of the computer software code and arranging the second SIP client to recognise the indicator.

12. A method as claimed in claim 1 which further comprises the step of proceeding with any SIP process related to the SIP message.

13. A method as claimed in claim 11 wherein said second SIP client is arranged such that on receipt of a SIP message containing such an indicator, the computer software code stored in the SIP message is executed by the second node before that second node carries out any other processes related to the SIP message.

14. A method as claimed in claim 1 wherein said computer software is arranged to interact with the second SIP client via a specified application programming interface.

15. A method as claimed in claim 1 wherein said computer software is arranged to interact with a processor associated with the second SIP client via a specified application programming interface.

16. A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to set up a multimedia conference call.

17. A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to upgrade said SIP client.

18. A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to carry out a self-test function and to pass the results to another node.

19. A method as claimed in claim 1 wherein a call is made to the first node, the first node is unable to accept the call and the first node sends the said SIP message and computer software code responsive to being unable to accept the call, said execution of said computer software code causes said second node to accept a forwarded call from the first node and to pass the identity of the second node back to a call originator.

20. A communications network node comprising:

(i) a SIP client;

(ii) an input arranged to receive SIP messages;

(iii) a processor arranged to extract and execute computer software code from a received SIP message.

21. A communications network node as claimed in claim 20 wherein said processor comprises a Java virtual machine.

22. A communications network node as claimed in claim 20 which further comprises an application programming interface arranged to allow the computer software code to interact with the SIP client.

23. A communications network node as claimed in claim 20 wherein said processor further comprises a detector arranged to detect an indicator in a received SIP message which indicates that computer software code is associated with that SIP message.

24. A computer program arranged to control a communications network node, said node comprising a SIP client and a processor, said computer program being

arranged to control the node when executed on the processor such that when a SIP message is received by the SIP client, which contains computer software code, the software code is executed by the processor.

25. A computer program as claimed in claim 24 which is stored on a computer readable medium.

26. A communications network comprising a plurality of communications network nodes each such node comprising:

(i) a SIP client;

(ii) an input arranged to receive SIP messages containing computer software code; and

(iii) a processor arranged such that in use, when a SIP message is received, any computer software code contained in that SIP message is executed by the processor.

27. (withdrawn) A method of setting up a conference call between two or more parties, each party comprising a SIP client and a host processor, said method comprising the steps of:

(i) storing computer software code in a SIP message;

(ii) sending the SIP message to each of the parties;

(iii) executing the computer software code at each of the host processors.

28. (withdrawn) A method as claimed in claim 27 wherein the computer software code is arranged to take into account capabilities of each host processor.

29. (withdrawn) A method as claimed in claim 27 wherein said conference call is a multimedia conference call.

30. (withdrawn) A system for automatically setting up a conference call between two or more parties, each party comprising a SIP client and a host processor,

said system comprising:- a processor for storing computer software code in a SIP message and to send that SIP message to each of the parties; and wherein each of said host processors is arranged to execute the computer software code in use, when the SIP message is received.

31. (withdrawn) A method of upgrading or replacing interconnected SIP clients each SIP client being associated with a host processor said method comprising the steps of:-

(i) storing computer software code suitable for said upgrade or replacement in a SIP message;

(ii) sending the SIP message to each of the SIP clients; and

(iii) executing the computer software at each of the host processors.

32. (withdrawn) A method of testing members of a group of SIP clients each SIP client being associated with a host processor said method comprising the steps of:-

(i) storing computer software code suitable for said testing in a SIP message;

(ii) sending the SIP message one of the SIP clients;

(ii) executing the computer software at the host processor associated with that SIP client in order to obtain test results; and

(iii) repeating steps (ii) to (iii) for each of the other SIP clients in the group.

33. (withdrawn) A method of forwarding a call from a first SIP client to a second SIP client, each of said SIP clients being associated with a host processor, said method comprising the steps of:-

(i) receiving a call at the first SIP client and if that call is not answered then storing computer software code in a SIP message, said computer software code being arranged to forward a call;

(ii) sending the SIP message from the first SIP client to a specified second SIP client; and

(iii) executing the computer software using the host processor associated with the second SIP client such that the call is forwarded to the second SIP client.

34. A method as claimed in claim 1 wherein said execution of said computer software code causes the second node to replace said SIP client.

Evidence Appendix

None.

Related Proceedings Appendix

None.